

# Wavelet Radiosity on Arbitrary Planar Surfaces

Nicolas Holzschuch<sup>1</sup>, François Cuny<sup>2</sup> and Laurent Alonso<sup>1</sup>

ISA research team

LORIA<sup>3</sup>

Campus Scientifique, BP 239

54506 Vandœuvre-les-Nancy CEDEX, France

**Abstract.** Wavelet radiosity is, by its nature, restricted to parallelograms or triangles. This paper presents an innovative technique enabling wavelet radiosity computations on planar surfaces of arbitrary shape, including concave contours or contours with holes. This technique replaces the need for triangulating such complicated shapes, greatly reducing the complexity of the wavelet radiosity algorithm and the computation time. It also gives a better approximation of the radiosity function, resulting in better visual results. Our technique works by separating the radiosity function from the surface geometry, extending the radiosity function defined on the original shape onto a simpler domain – a parallelogram – better behaved for hierarchical refinement and wavelet computations.

## 1 Introduction

Wavelet radiosity [12] is one of the most interesting technique for global illumination simulation. Recent research [7] has shown that higher order multi-wavelets ( $\mathcal{M}_2$  and  $\mathcal{M}_3$ ) are providing a very powerful tool for radiosity computations. Multi-wavelets can approximate the radiosity function efficiently with a small number of coefficients. As a consequence, they give a solution of better quality in a shorter time.

Multi-wavelets are defined only on parallelograms and triangles. This causes problems for radiosity computations on scenes coming from real world applications, such as architectural scenes, or CAD scenes. In such scenes, planar surfaces have a fairly complicated shape (see figure 1 and 12(a)). To do wavelet radiosity computations on such scenes, we have to tessellate these planar shapes into triangles and parallelograms, which results in a large number of input primitives (see figure 1(b)). Furthermore, this decomposition is purely geometrical and was not based on the illumination, yet it will influence our approximation of the radiosity function. In some cases, this geometric decomposition results in a poor illumination solution (see figure 2(a) and 11(a)).

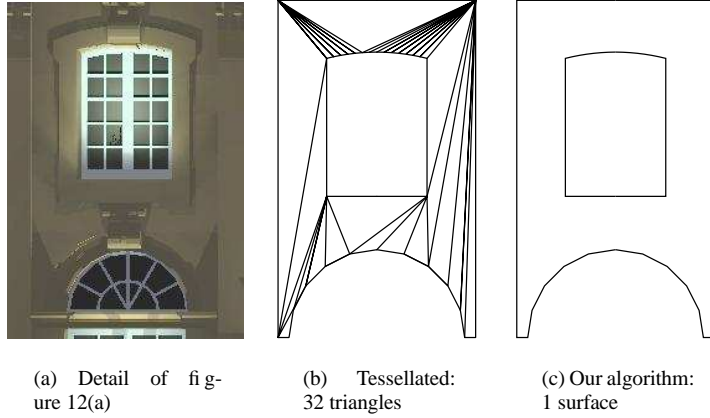
In the present paper, we separate the radiosity function from the surface geometry. This enables us to exploit the strong approximating power of multi-wavelets for radiosity computations, with planar surfaces of arbitrary shape – including concave contours, contours with holes or disjoint contours. Our algorithm results in a better approximation of the radiosity function (see figure 2(b) and 11(b)) with a smaller number of input primitives, faster convergence and lower memory costs.

---

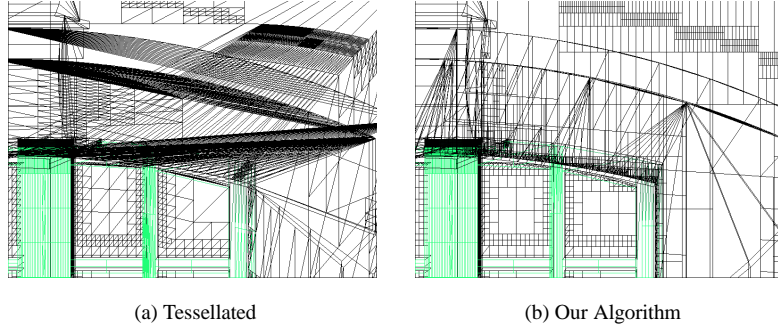
<sup>1</sup>INRIA Lorraine.

<sup>2</sup>Institut National Polytechnique de Lorraine.

<sup>3</sup>UMR n° 7503 LORIA, a joint research laboratory between CNRS, Institut National Polytechnique de Lorraine, INRIA, Université Henri Poincaré and Université Nancy 2.



**Fig. 1.** Planar surfaces can have a fairly complicated shape



**Fig. 2.** Wavelet radiosity on arbitrary planar surfaces (see also figure 11)

Our algorithm extends the radiosity function defined on the original shape onto a simpler domain, better behaved for hierarchical refinement and wavelet computations. This extension of the radiosity function is defined to be easily and efficiently approximated by multi-wavelets. The wavelet radiosity algorithm is modified to work with this abstract representation of the radiosity function.

Our paper is organised as follows: in section 2, we will review previous work on radiosity with planar surfaces of complicated shape. Section 3 is a detailed explanation of our algorithm and of the modifications we brought to the wavelet radiosity algorithm. Section 4 presents the experiments we have conducted with our algorithm on different test scenes. Finally, section 5 presents our conclusions.

## 2 Previous work

The wavelet radiosity method was introduced by [12]. It is an extension of the radiosity method [11] and especially of the hierarchical radiosity method [13]. It allows the use of higher order basis functions in hierarchical radiosity.

In theory, higher order wavelets are a very powerful tool to approximate rapidly varying functions with little coefficients. In practice, they have several drawbacks, especially in terms of memory costs. In the early implementations of wavelets bases in the radiosity algorithm, these negative points were overcoming the positive theoretical advantages [19]. Recent research [7] has shown that using new implementation methods [2, 3, 7, 18, 21] we can actually exploit the power of higher order wavelets, and that their positive points are now largely overcoming the practical problems. They provide a better approximation of the radiosity function, with a small number of coefficients, resulting in faster convergence and smaller memory costs.

On the other hand, higher order wavelets, and especially multi-wavelets ( $\mathcal{M}_2$  and  $\mathcal{M}_3$ ) are defined as the tensor products of one-dimensional wavelets. As a consequence, they are defined over a square. The definition can easily be extended on parallelograms or triangles, but higher order wavelets are not designed to describe the radiosity function over complex surfaces.

Such complex surfaces can occur in the scenes on which we do global illumination simulations. Especially, scenes constructed using CAD tools such as CSG geometry or extrusion frequently contain complex planar surfaces, with curved boundaries or holes in them.

The simplest solution to do radiosity computations on such surfaces is to tessellate them into triangles, and to do radiosity computations on the result of the tessellation. This method has several negative consequences on the radiosity algorithm:

- It increases the number of input surfaces and the algorithmic complexity of the radiosity algorithm is linked to the square of the number of input surfaces.
- The tessellation is made before the radiosity computations and it influences these computations. It can prevent us from reaching a good illumination solution.
- The tessellation does not allow a hierarchical treatment over the original surface, only over each triangle created by the tessellation. We can not fully exploit the capabilities of hierarchical radiosity, and especially of wavelet radiosity.
- By artificially subdividing an input surface into several smaller surfaces, we are creating discontinuities. These discontinuities will have to be treated at some point in the algorithm.
- Tessellation can create poorly shaped triangles (see figure 1(b)), or slivers. These slivers can cause Z-buffer artifacts when we visualise the radiosity solution, and are harder to detect in visibility tests (*e.g.* ray-casting).

Some of these problems can be removed by using clustering [10, 16, 17]. In clustering, neighbouring patches are grouped together, into a *cluster*. The cluster receives radiosity and distributes it to the patches that it contains. On the other hand, current clustering strategies are behaving poorly in scenes with many small patches located close to each other [14]. It would probably be more efficient to apply clustering to the original planar surfaces instead of applying it to the result of the tessellation.

A better grouping strategy is face-clustering [20]. In face-clustering, neighbouring patches are grouped together according to their coplanarity. Yet even face-clustering depends on the geometry created by the tessellation. Furthermore, it would not allow us to exploit the strong approximating power of multi-wavelets.

- if the original planar shape is polygonal:
  - compute its convex hull (in linear time) using the chain of points [9].
  - compute the minimal enclosing parallelograms of the convex hull (in linear time) using Schwarz *et al.* [15].
  - if the previous algorithm gives several enclosing parallelograms, select the one that has angles closer to  $\frac{\pi}{2}$ .
- if the original shape is a curve, or contains curves:
  - approximate the curve by a polygon
  - compute the enclosing parallelogram of the polygon
  - compute the extrema of the curve in the directions of the parallelogram.
  - if needed, extend the parallelogram to include these extrema.

**Fig. 3.** Our algorithm for finding an enclosing parallelogram.

Bouatouch *et al.* [5] designed a method for discontinuity meshing and multi-wavelets. In effect, they are doing multi-wavelets computations over a non-square domain. However, their algorithm requires several expensive computations of push-pull coefficients. Our algorithm avoids these computations.

Baum *et al.* [1] designed a method for radiosity computations with arbitrary planar polygons, including polygons with holes. Their method ensures that the triangles produced are well-shaped, and suited for radiosity computations. Since it is designed for non-hierarchical radiosity, it is done in a preliminary step, before all radiosity computations. Our method, designed for wavelet radiosity, acts during the global illumination simulation, and adapts the refinement to the radiosity.

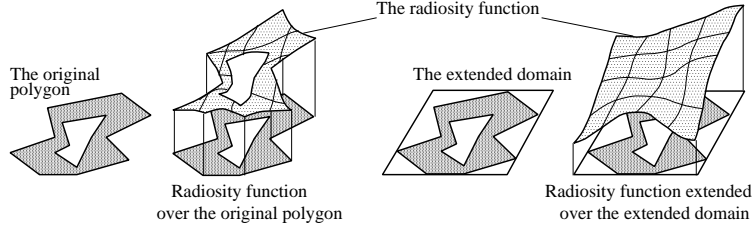
### 3 The Extended Domain Algorithm

In this section, we present our algorithm for wavelet radiosity computations on planar surfaces of arbitrary shape. Our algorithm separates the radiosity function from the surface geometry; we introduce a simple domain that will be used for radiosity computations. The radiosity function on the original surface is inferred from the radiosity on the simple domain.

Section 3.1 explains how we select an extended domain for our computations. In section 3.2, we describe how we extend the definition of the radiosity function over this domain. The extended domain is then used in a wavelet radiosity algorithm like an ordinary patch, with some specific adjustments. These adjustments are described in section 3.3.

#### 3.1 Selection of an extended domain

The first step of our algorithm is the choice of an extended domain, which we will use for wavelet radiosity computations. This extended domain must obey two rules:



**Fig. 4.** Extending the radiosity function over the extended domain.

- it must enclose the original shape,
- it must be well suited for wavelet radiosity computations.

Since multi-wavelets ( $\mathcal{M}_2, \mathcal{M}_3 \dots$ ) are defined as tensor products of one-dimensional wavelets, the second rule implies that the extended domain must be a parallelogram. Moreover, if this parallelogram is closer to a rectangle, there will be less distortions in the wavelet bases, resulting in a better approximation of the radiosity function. So we want the angles of the parallelogram to be close to  $\frac{\pi}{2}$ .

Since only radiosity computations made on the original shape are of interest, we also want the enclosing parallelogram to be as close as possible from the original shape.

Basically, any parallelogram satisfying these criterions could be used with our algorithm. The algorithm used in our implementation is described in figure 3. The key point is that this algorithm runs in linear time with respect to the number of vertices: the convex hull of a chain of points in 2D can be computed in linear time [9], and Schwarz's algorithm for the enclosing parallelogram is also linear [15].

### 3.2 Extending the radiosity function over this domain

Once we have an extended domain, we need to define the radiosity function over this domain. This extension of the radiosity function must obey two rules (see figure 4):

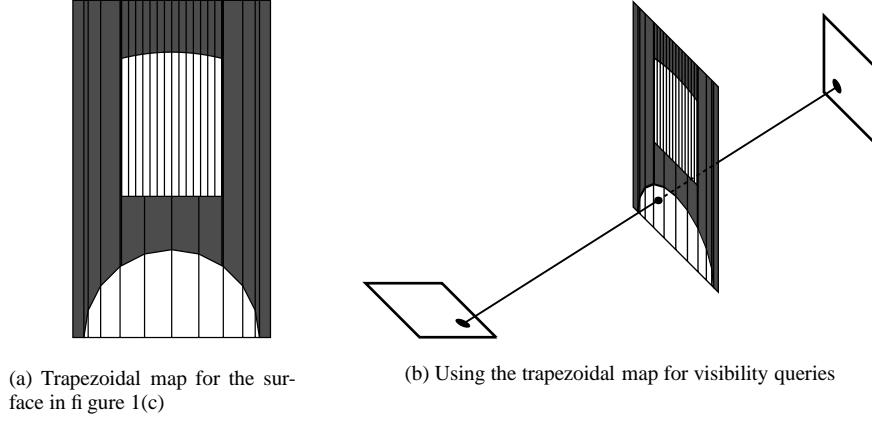
- it must be equal to the original radiosity function over the original domain
- it must be as simple and as smooth as possible, to be efficiently approximated by multi-wavelets.

The second point is crucial: we have to compute the radiosity function over the entire domain. Because of the hierarchical nature of wavelets, during the push-pull step radiosity values computed at one point of the domain can influence other points of the domain. So our extension of the radiosity function must be computed with the same precision regardless of whether we are on the original surface or not.

Since the discontinuities of the radiosity function and its derivatives only come from visibility discontinuities, we do not want to introduce more visibility discontinuities in our extension. We define an *extended visibility* function  $V'$ : the visibility between a point  $Q$  in space and a point  $P$  on the extended domain is defined as the visibility between  $Q$  and  $P'$ , where  $P'$  is defined as the closest point from  $P$  on the original planar surface:

$$V'(Q, P) = V(Q, P')$$

Of course, if  $P$  is already on the original planar surface,  $P'$  is equal to  $P$ . In that case, the extended visibility function is equal to the standard visibility function.



**Fig. 5.** Trapezoidal map of an arrangement of line segments

The radiosity function on the extended domain is then defined as the radiosity function, as computed by the wavelet radiosity algorithm, using this extended visibility function in the radiosity kernel.

### 3.3 Using the extended domain in the wavelet radiosity algorithm

In this section, we describe our adaptation of the wavelet radiosity algorithm to work with our extended domains. We use a standard wavelet radiosity algorithm [7, 21]. The core of the algorithm is left unchanged (refinement oracle, link storage). We will review here the points that require some special attention:

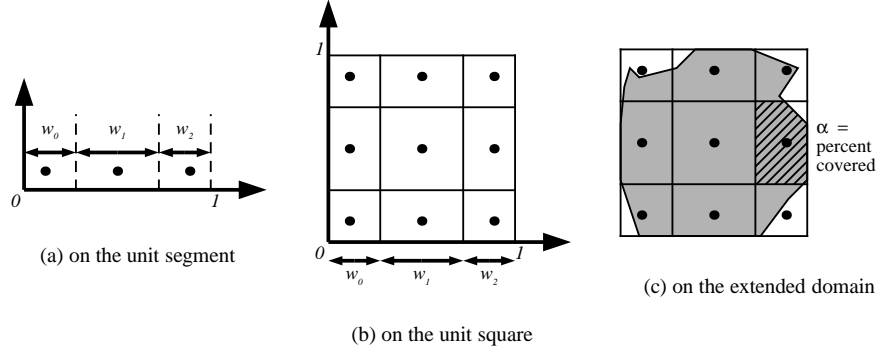
- reception and push-pull
- visibility
- emission
- refinement

**Reception and Push-Pull.** The wavelet radiosity algorithm is a hierarchical algorithm. During the push-pull step, radiosity values computed at one point of the patch can influence the representation of radiosity for the entire patch. Hence, we want the same precision for all radiosity computations over the entire patch.

For reception, the extended domain is therefore treated just like an ordinary patch. All parts of the extended domain are receiving radiosity, with the same precision, regardless of whether or not they belong to the original planar surface.

Similarly, we are doing the push-pull step over the entire extended domain, without any reference to the original surface. Since our extended domain is by design a parallelogram, instead of an ordinary polygon, we do not have to compute any expensive push-pull coefficients.

**Visibility.** For all the visibility computations, only the original planar surface can act as an occluder. The extended domain is never used in visibility computations.



**Fig. 6.** The weights of the quadrature points can be seen as the area of a *zone of influence*.

To detect if the original surface is actually occluding an interaction, we compute the trapezoidal map of an arrangement of line segments [4, 8] over the segments of the contour of the original surface (see figure 5). For each trapeze, we store its status – whether it is inside or outside of the original surface.

Using randomized algorithms, trapezoidal maps can be constructed in time  $O(n \log n)$ , where  $n$  is the number of vertices. Once constructed, they can be queried in  $O(\log n)$  time. Since the construction algorithm is randomised, we shuffle the segments of the original surface before building the trapezoidal map.

Visibility queries in our radiosity algorithm are visibility queries between two points, either two quadrature points [7] or the closest point on the original surface from a quadrature point (see section 3.2). We compute the intersection between the ray joining these quadrature points and the supporting plane of the original surface, check whether the intersection point is inside the bounding box of the extended domain, then check whether it is inside the extended domain itself, then query the trapezoidal map to check if it is inside or outside the original surface.

**Emission.** During the reception, the entire extended domain has received illumination. The radiosity received over parts of the extended domain that are not included in the original surface does not exist in reality, and it should not be sent back into the scene. Otherwise, there would be an artificial creation of energy, violating the principle of conservation of energy.

Because of the hierarchical nature of the wavelet radiosity algorithm, it would be difficult to compute the exact part of this radiosity function that really exists. Instead, we act on the weights of the quadrature points.

In the wavelet radiosity algorithm, all the transfer coefficients between an emitter and a receiver are computed using quadratures. Quadratures allow the evaluation of a complex integral by sampling the function being integrated at the quadrature points, and multiplying the values by quadrature weights. Most implementations use Legendre-Gauss quadratures.

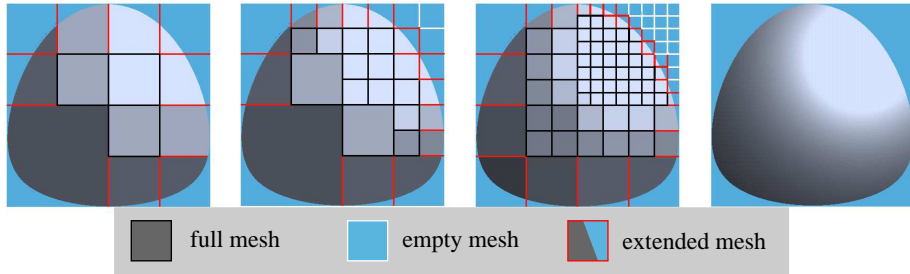
Since the weights are positive and their sum is equal to 1, you can visualise them as being the length of a *zone of influence* for the corresponding quadrature point (see figure 6(a) for the one dimension case). The same applies in two dimensions: the weights of the quadrature points can be seen as the area of a zone of influence (see

```

for each interaction  $s \rightarrow r$ :
  for each quadrature point  $q_i$  on the emitter  $s$ 
     $A_i =$  area of influence of  $q_i$ 
     $\alpha_i \leftarrow$  percentage of  $A_i$  that is inside the original emitter
     $q'_i =$  nearest point from  $q_i$  on the emitter
    for each quadrature point  $p_j$  on the receiver  $r$ 
       $p'_j =$  nearest point on the receiver
       $V(q'_i, p'_j) =$  visibility between  $q'_i$  and  $p'_j$ 
       $G(q_i, p_j) =$  radiosity kernel between  $q_i$  and  $p_j$ 
       $B_r += \alpha_i w_i w_j B_s(q_i) V(q'_i, p'_j) G(q_i, p_j)$ 
    end for
  end for

```

**Fig. 7.** Pseudo-code for wavelet radiosity emission using the extended domain.



**Fig. 8.** Refinement of the extended domain

figure 6(b)); the weight of quadrature point  $p_{i,j}$  is  $w_i w_j$ . Please note that these zones of influence are not equal to the Voronoï diagram of the quadrature points.

We suggest an extension to the Gaussian quadrature to take into account the fact that the extended domain is not entirely covered by the actual emitter: the weight of a quadrature point is multiplied by the proportion of its area of influence that is actually covered by the emitter. For example, on figure 6(c), the weight of the quadrature point in the hashed area should be  $w_1 w_2$ . Since the fraction of its area of influence covered by the emitter is  $\alpha$ , the weight used in the computation will be  $\alpha w_1 w_2$ .

Our method allows for a quick treatment of low precision interactions, and for high precision interactions, it tends toward the exact value. The more we refine an interaction, the more precision we get on the radiosity on the emitter. We also get the exact value if the zone of influence is entirely full or entirely empty.

In some cases, it can happen that the quadrature point falls outside the original emitter. We use these quadrature points anyway.

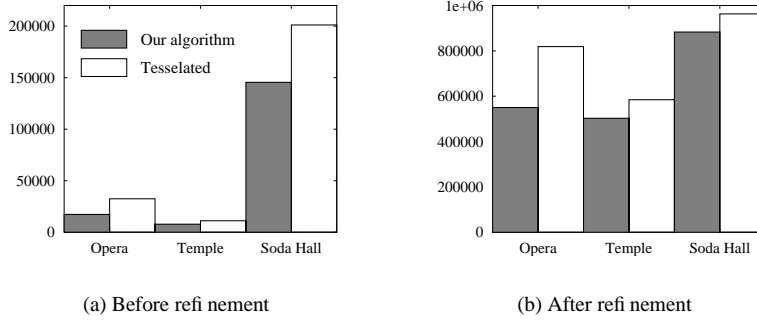
Figure 7 shows the pseudo-code for radiosity emission using the extended domain.

**Refinement.** As with the original wavelet radiosity algorithm, the extended domain can be subdivided if the interaction needs to be subdivided. The refinement oracle deals with the extended domain as it would deal with any other patch. Because of the hierarchical representation of the radiosity function in the wavelet radiosity algorithm, we must have the same precision on the radiosity function over the entire domain. The push-pull step can make parts of the domain that are not inside the original surface influence our representation of the radiosity function over the entire domain.



Name	# initial surfaces	after tessellation	ratio
Opera	17272	32429	1.88
Temple	7778	11087	1.43
Soda Hall	145454	201098	1.38

**Table 1.** Description of our test scenes



**Fig. 9.** Number of patches in our test scenes

If the extended domain is refined, we deal with each part of the subdivided extended domain as we would deal with the original extended domain. Two special cases can appear (see figure 8):

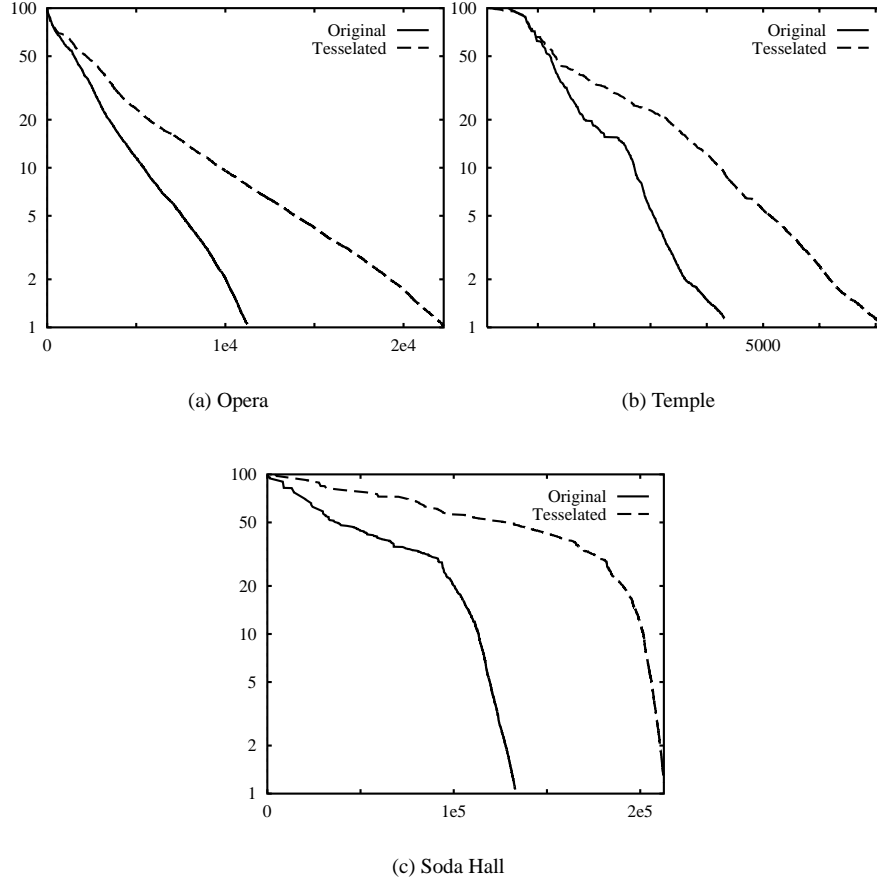
- if the result of the subdivision does not intersect at all with the original planar surface, it is empty. Therefore it cannot play a role in the emission of radiosity, but we keep computing the radiosity function over this patch.
- if the result of the subdivision is totally included inside the original planar surface. In that case, we are back to the standard wavelet radiosity algorithm on parallelograms.

## 4 Experiments

We have tested our algorithm for wavelet radiosity on arbitrary planar surfaces on various test scenes (see figure 12 for images of our test scenes, and table 1 and figure 9(a) for their description). We were interested in a comparison between our algorithm and the standard wavelet radiosity algorithm, acting on parallelograms and triangles. All the computations were conducted on the same computer, a SGI Origin 2000, using a parallel version [6] of our wavelet radiosity algorithm [7, 21].

In all these test scenes, the number of surfaces after tessellation is less than twice the number of surfaces in the original scene. Much less than what could be expected from figure 1. Most of the initial surfaces in the scenes are parallelograms or triangles, and don't require tessellation.

The first result is that our algorithm gives better visual quality than doing wavelet radiosity computations on a tessellated surface (see figure 2 and 11). Our separation of the radiosity function from the surface geometry results in a better approximation of the radiosity function.



**Fig. 10.** Convergence rate (un-shot energy over initial energy) as a function of computation time (in seconds).

Beyond this important result, we were interested in a comparison of computation time and memory costs for both algorithms.

Obviously, our algorithm reduces the number of patches, and therefore the memory cost of the initial scene (see table 1 and figure 9(a)). According to our computations, it also reduces the number of patches in the final scene, although not in the same proportions (see figure 9(b)).

The later result was to be expected: the wavelet radiosity algorithm will refine the original scene a lot, resulting in numerous sub-patches. The number of patches in the scene after the radiosity computations is mainly linked to the complexity of the radiosity function itself, and not to the complexity of the scene. However, it appears that our algorithm results in more efficient refinement, since we reach convergence with a smaller number of patches. In some scenes, we can reach convergence with 30 % less patches.

The fact that our algorithm allows for more efficient refinement also appears in the

computation times (see figure 10). In our experiments, we measure the energy initially present in the scene and the energy that hasn't yet been propagated in the scene. The ratio of these two measures tells us how far we are from complete convergence. Figure 10 displays this ratio as a function of the computation time, both for our algorithm and for the wavelet radiosity algorithm operating on a tessellated version of the scene. Our algorithm ensures a faster convergence on all our test scenes. The speedup is of about 30 %, which shows that acting on the original planar surface instead of the tessellated surface gives more efficient refinement.

## 5 Conclusion

In conclusion, we have presented a method to separate the radiosity function from the surface geometry. This method removes the need to tessellate complex planar surfaces, resulting in a more efficient global illumination simulation, with better visual quality. Our method results in faster convergence, with smaller memory costs.

In our future work, we want to extend this algorithm to discontinuity meshing. Discontinuity meshing introduces a geometric model of the discontinuities of the radiosity function and its derivatives, the *discontinuity mesh*. The discontinuity mesh provides optimal meshing for radiosity computations near the discontinuities. The discontinuity mesh is a complicated structure, and it can influence radiosity computations away from the discontinuities, for example because of triangulation. We want to use our algorithm to smoothly integrate the discontinuity mesh in the natural subdivision for multi-wavelet radiosity, removing the need to tessellate the discontinuity mesh.

We also want to explore a combination of our algorithm with clustering techniques. First, our algorithm could be used to group together neighbouring coplanar patches in a natural way. This would help the clustering strategy [14] and give a more accurate result. Second, we would like to integrate our algorithm with face-clustering, bringing multi-wavelets into face-clusters.

Finally, our separation of the radiosity function from the surface geometry could also be used to compute radiosity using multi-wavelets on curved surfaces. There are several parametric surfaces for which the limits of the parametric space are not square. We suggest using our algorithm to enclose these limits into a square limit, making it easier for multi-wavelets.

## 6 Acknowledgements

Permission to use the Soda Hall model<sup>4</sup> was kindly given by Prof. Carlo Sequin.

Jean-Claude Paul has started and motivated all this research. The authors would like to thank him for his kind direction, support and encouragements.

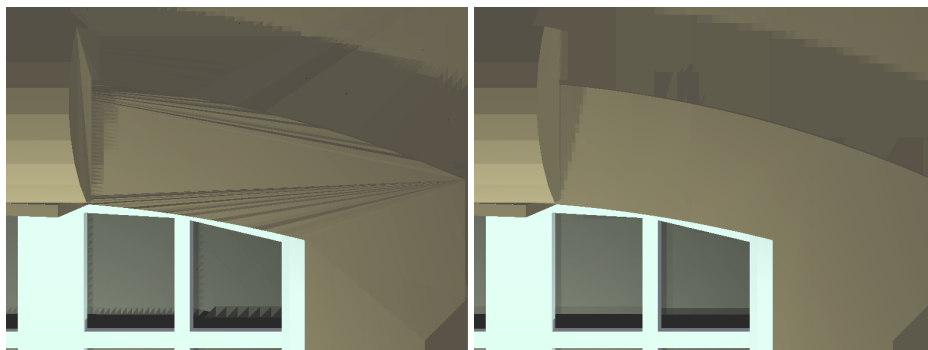
## References

1. D. R. Baum, S. Mann, K. P. Smith, and J. M. Winget. Making Radiosity Usable: Automatic Preprocessing and Meshing Techniques for the Generation of Accurate Radiosity Solutions. *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, 25(4):51–60, July 1991.
2. P. Bekaert and Y. Willems. Error Control for Radiosity. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 153–164, New York, NY, 1996. Springer-Verlag/Wien.

---

<sup>4</sup>The Soda Hall model is available on the web, at <http://www.cs.berkeley.edu/~kofler>.

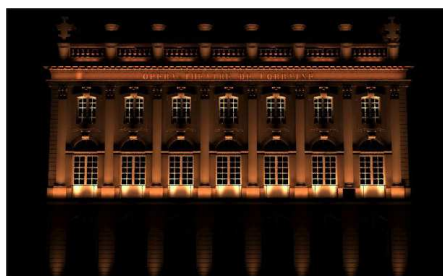
3. P. Bekaert and Y. D. Willems. Hirad: A Hierarchical Higher Order Radiosity Implementation. In *Proceedings of the Twelfth Spring Conference on Computer Graphics (SCCG '96)*, Bratislava, Slovakia, June 1996. Comenius University Press.
4. J.-D. Boissonnat and M. Yvinec. *Algorithmic Geometry*. Cambridge University Press, 1998.
5. K. Bouatouch and S. N. Pattanaik. Discontinuity Meshing and Hierarchical Multiwavelet Radiosity. In W. A. Davis and P. Prusinkiewicz, editors, *Proceedings of Graphics Interface '95*, pages 109–115, San Francisco, CA, May 1995. Morgan Kaufmann.
6. X. Cavin, L. Alonso, and J.-C. Paul. Parallel Wavelet Radiosity. In *Second Eurographics Workshop on Parallel Graphics and Visualisation*, pages 61–75, Rennes, France, Sept. 1998.
7. F. Cuny, L. Alonso, and N. Holzschuch. A novel approach makes higher order wavelets really efficient for radiosity. *Computer Graphics Forum (Eurographics 2000 Proceedings)*, 19(3), Sept. 2000. To appear. Available from <http://www.loria.fr/~holzschu/Publications/paper20.pdf>.
8. O. Devillers, M. Teillaud, and M. Yvinec. Dynamic location in an arrangement of line segments in the plane. *Algorithms Review*, 2(3):89–103, 1992.
9. H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Nov. 1987.
10. S. Gibson and R. J. Hubbard. Efficient hierarchical refinement and clustering for radiosity in complex environments. *Computer Graphics Forum*, 15(5):297–310, Dec. 1996.
11. C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. Modelling the Interaction of Light Between Diffuse Surfaces. *Computer Graphics (ACM SIGGRAPH '84 Proceedings)*, 18(3):212–222, July 1984.
12. S. J. Gortler, P. Schroder, M. F. Cohen, and P. Hanrahan. Wavelet Radiosity. In *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, pages 221–230, 1993.
13. P. Hanrahan, D. Salzman, and L. Aupperle. A Rapid Hierarchical Radiosity Algorithm. *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, 25(4):197–206, July 1991.
14. J. M. Hasenfratz, C. Domez, F. Sillion, and G. Drettakis. A practical analysis of clustering strategies for hierarchical radiosity. *Computer Graphics Forum (Eurographics '99 Proceedings)*, 18(3):C–221–C–232, Sept. 1999.
15. C. Schwarz, J. Teich, A. Vainshtein, E. Welzl, and B. L. Evans. Minimal enclosing parallelogram with application. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages C34–C35, 1995.
16. F. Sillion. A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3), Sept. 1995.
17. B. Smits, J. Arvo, and D. Greenberg. A Clustering Algorithm for Radiosity in Complex Environments. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 435–442, 1994.
18. M. Stamminger, H. Schirmacher, P. Slusallek, and H.-P. Seidel. Getting rid of links in hierarchical radiosity. *Computer Graphics Journal (Proc. Eurographics '98)*, 17(3):C165–C174, Sept. 1998.
19. A. Willmott and P. Heckbert. An empirical comparison of progressive and wavelet radiosity. In J. Dorsey and P. Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 175–186, New York, NY, 1997. Springer Wien. ISBN 3-211-83001-4.
20. A. Willmott, P. Heckbert, and M. Garland. Face cluster radiosity. In *Rendering Techniques '99*, pages 293–304, New York, NY, 1999. Springer Wien.
21. C. Winkler. *Expérimentation d'algorithmes de calcul de radiosit     base d'ondelettes*. Th     d'universit    , Institut National Polytechnique de Lorraine, 1998.



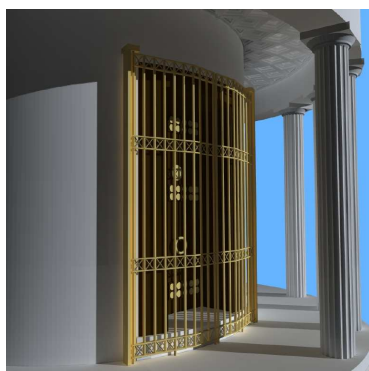
(a) Tessellated

(b) Our Algorithm

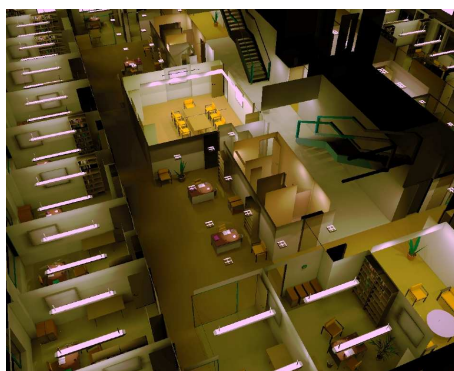
**Fig. 11.** Using our algorithm for wavelet radiosity on arbitrary planar surfaces (see also figure 2)



(a) Opera



(b) Temple



(c) Soda Hall

**Fig. 12.** Our test scenes